DOCUMENT RESUME

| | | |
|---|---|---|
| ED 442 823 | | TM 031 243 |

| | |
|---|---|
| AUTHOR | Luecht, Richard M. |
| TITLE | Implementing the Computer-Adaptive Sequential Testing (CAST) Framework To Mass Produce High Quality Computer-Adaptive and Mastery Tests. |
| PUB DATE | 2000-04-00 |
| NOTE | 42p.; Paper presented at the Annual Meeting of the National Council on Measurement in Education (New Orleans, LA, April 25-27, 2000). |
| PUB TYPE | Reports - Descriptive (141) -- Speeches/Meeting Papers (150) |
| EDRS PRICE | MF01/PC02 Plus Postage. |
| DESCRIPTORS | *Adaptive Testing; *Computer Assisted Testing; *Mastery Tests; *Test Construction |
| IDENTIFIERS | Large Scale Assessment; *Sequential Testing; Testlets |

ABSTRACT

Computerized testing has created new challenges for the production and administration of test forms. This paper describes a multi-stage, testlet-based framework for test design, assembly, and administration called computer-adaptive sequential testing (CAST). CAST is a structured testing approach that is amenable to both adaptive and mastery testing. Four aspects of CAST are discussed: (1) designing CAST test targets and specifications; (2) using automated test assembly to build the CAST forms; (3) security controls in CAST; and (4) large-scale data management considerations. (Contains 22 references.) (Author/SLD)

# Implementing the Computer-Adaptive Sequential Testing (CAST) Framework to Mass Produce High Quality Computer-Adaptive and Mastery Tests

**Richard M. Luecht**

**University of North Carolina at Greensboro**

**April, 2000**



Paper presented in the National Council on Measurement in Education Symposium, N3, *The Development and Application of Multi-Stage, Testlet Based Designs for Computerized Adaptive Testing*, New Orlean, LA, April 25-27, 2000.

**Abstract**

Computerized testing has created new challenges for the production and administration of test forms. This paper describes a multi-stage, testlet-based framework for test design, assembly and administration called *computer-adaptive sequential testing* (CAST). CAST is a structured testing approach that is amenable to both adaptive and mastery testing. Four aspects of CAST are discussed in this paper: (1) designing CAST test targets and specifications, (2) using automated test assembly to build the CAST forms, (3) security controls in CAST, and (4) large-scale data management considerations.

## Introduction

Computer-adaptive sequential testing (CAST) was developed as a integrated framework for high-stakes multi-stage computer-adaptive and mastery tests (Luecht, Nungester, & Hadadi, 1996; Luecht & Nungester, 1998). This paper provides an overview of CAST in the context of multi-stage adaptive testing, although, extensions to multi-stage, sequential mastery testing are also possible (Luecht & Nungester, 1998). Furthermore, this paper explores four important aspects of CAST: (1) strategies for designing CAST forms; (2) using automated test assembly to build CAST forms; (3) security issues under CAST; and (4) large-scale data management issues.

## The CAST Framework

When Luecht and Nungester (1998) generated the *CAST framework*, they introduced some new, innocuous terminology such as modules, panels and pathways. Their intent was not to create new jargon, but rather, to avoid some of the messy connotations associated with concepts such as testlets, staging tests, and test forms. For similar reasons, I will use with their original descriptors here, with some latitude. There are four basic test design/administration concepts in CAST: (1) modules, (2) panels, (3) stages, and (4) pathways.

*Modules* and *panels* are the two basic "units" in CAST. *Modules* are the building block units in CAST. They are groups of items or performance tasks that are somewhat homogeneous with respect to item difficulty and which are administered as a unit, with presentation order randomized or fixed or both. In the current vernacular of multi-

stage testing, modules can be thought of as "testlets" (cf. Wainer and Kiely, 1987), although, the testlet concept can be ambiguous in some contexts. Like testlets, modules may be linked by some central theme (e.g., content, a case vignette, or a reading passage), however, there is no requirement in CAST to do so. In fact, modules can comprise several item sets and associated case vignettes. Each module can be constructed according to its unique specifications for content and statistical characteristics. Or, the union of several modules may satisfy a more global set of test specifications. In the latter instance, the inter-relationships among the statistical and qualitative characteristics of the modules may be important.

Items can be assigned to multiple modules, depending on the rules covering reuse and item exposure. The size of the modules can range from small (five to ten items) to large (50 to 100 items), depending on the nature of the test. Modules can also vary in size across stages and by average difficulty.

A specified number of modules or testlets are assigned to what Luecht and Nungester called *panels*. A panel is the basic organizing unit in CAST from the perspectives of test design, assembly and administration. Multiple panels can be constructed, numbered and administered, just like test forms. The major difference between panels and test forms is that panels have their own "administration rules" and can produce either an adaptive test or a mastery test, or a hybrid of both. Panels consolidate the modules in distinct ways and facilitate data management at many levels.

Within each panel, the modules or testlets are assigned to designated test administration *stages*, providing the "multi-stage" aspect of CAST. The panels can be flexibly configured to have two or more stages and any number of modules per stage. Practically speaking, CAST panels would rarely have more than five stages.

In general, adding more stages and using smaller modules will increase the adaptive flexibility of the panels; however, empirical work at the National Board of Medical Examiners has demonstrated that, for long tests, using fewer stages and designing larger modules may be adequate from a psychometric perspective of score precision and preferable from examinees' perspectives in terms of allowable item review and perceptual changes in difficulty when transitioning between modules (Luecht, Nungester, Swanson & Hadadi, 1998; Luecht & Nungester, 1998). Note that all panels for a given testing program will obviously have the same number of stages. Within a stage, all assigned modules must be of equal size. Across stages, modules can vary in size (see Luecht & Nungester, 1998).

Test administration routing rules must be developed to explicitly control which modules are administered to different examinees at each stage of testing. The routing rules function akin to standard adaptive algorithms insofar as sequentially optimizing the selection decisions about which module to administer at each stage. The various routes that examinees can follow from module-to-module or testlet-to-testlet within the panel are called *pathways*. ***These pathways are critical in terms of test design, test assembly, quality assurance and test administration.***

Figure 1 displays a single panel for a four-stage adaptive test with ten modules or testlets. The stages go from bottom to top. Each module is pre-assigned to one of the four stages.

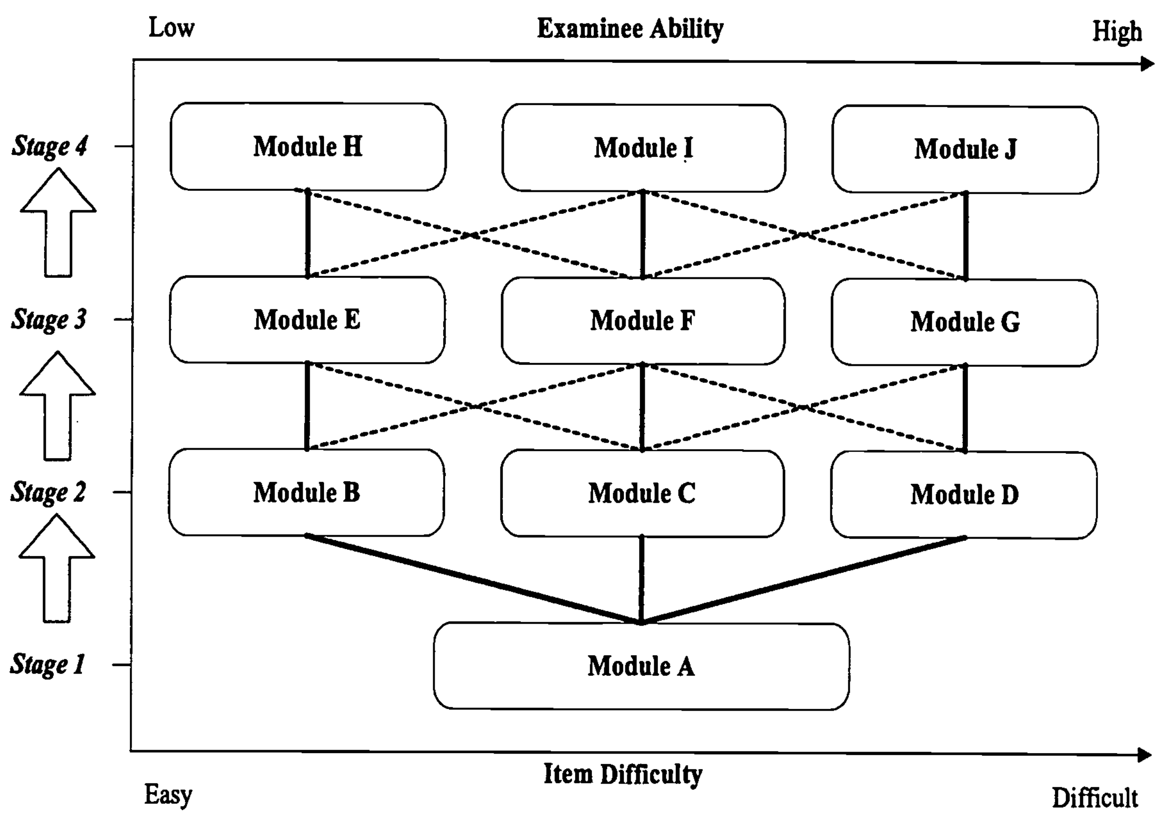


Figure 1. A CAST Panel with Four Stages, Ten Modules, and Three Primary Pathways

Module A is assigned to Stage 1. Modules B, C and D are assigned to Stage 2. Modules E, F, and G are assigned to Stage 3 and Modules H, I, and J are assigned to Stage 4. As indicated on the lower "item difficulty" scale, Modules B, E, and H are *easy* modules, targeted for lower proficiency examinees (see upper "examinee ability" scale).

Modules A, C, F, and I are *moderately difficult* modules. Modules D, G, and J are hard modules, targeted for high proficiency examinees.

The panel in Figure 1 has three primary *pathways*, each indicated by a solid connector line between the modules or testlets: (1) an *easy* pathway (A+B+E+H) for lower proficiency examinees; (2) a moderate pathway (A+C+F+I); and a hard pathway (A+D+G+J). The *secondary* pathways are denoted by the dashed lines (e.g., A+B+F+I). These secondary pathways (dashed connector lines) are completely under the control of the test developer and can be used to preclude certain pathways as a matter of test administration policy. For example, notice that there is no pathway from Module B to Module G.

Under CAST, an examinee is assigned to take a *panel*[1] instead of a test form. Realize that there could be several or hundreds of "active" panels constructed as a security measure against cheating. The panel assignments can involve pre-determined decisions based upon on retake policy rules and other criteria, or, may use a real-time random assignment algorithm to select a panel from a "panel pool" (i.e., active panels).

During test delivery, the first module is administered (for example, Module A in Figure 1). The items in each module may be administered sequentially, although

---

[1]In the terminology of modern object-oriented database systems and programming, *panels* are true "objects" that essentially "know" how to administer and score themselves. That is, panels are *encapsulated* test assembly objects. Although an in-depth technical discussion of the object-oriented nature and advantages of CAST panels is beyond the scope of this paper, I will allude to some of the salient advantages of "panels-as-objects" throughout this paper.

randomized presentation is certainly preferable for security reasons as well as to minimize contextual interdependencies among the patterns of item responses. The module may have its own time limits or an overall test administration timer may be running in the background. While the examinee is completing the module or testlet, (s)he can [normally] review and change answers–that is, there is no substantive or technical reason to prevent examinees from doing this in CAST. When the examinee has answered all of the questions in the module/testlet, or when time expires, a provisional score is computed (i.e., a weighted or unweighted number-correct score or an IRT-based proficiency score). The routing decisions and the routing rules are engaged to optimally select one of the modules or testlets from the next stage[2].

### Designing CAST Panels

There are two fundamental issues that largely determine how one goes about designing a CAST panel configuration. First, there is the issue of test score precision (i.e. test information). Where is the precision needed? Second, there is the issue of the auxiliary quantitative and qualitative test specifications (e.g., content, item types, word counts, and cumulative average time per item). That is, can the qualitative and non-

---

[2]In the context of a sequential mastery test, the routing procedures can include sophisticated statistical decision-making techniques like the sequential probability ratio test (Wald, 1947). Furthermore, the concept of pathways can even be exploited to route clearly failing examinees to a "diagnostic" set of modules useful for computing reliable diagnostic scores to highlight the examinees' strengths and weaknesses, by merely "turning off" the appropriate secondary pathways.

statistical quantitative test specifications be broken down and specified at the module or testlet level, OR, is it necessary to more holistically consider the test-level specifications for *combinations* of modules?

The score precision issue is largely a straight-forward psychometric issue that involves generating target test information functions for use in automated test assembly. The latter test specifications issue is the more tedious one. For many high-stakes testing programs, content and other important test features simply cannot be specified at the module level. For example, Hadadi, Swanson & Luecht (1999) demonstrated a real-life test design problem using automated test assembly (ATA) for the United States Medical Licensing Examination™ Step 1 (Federation of State Medical Board and the National Board of Medical Examiners) that employed almost 5,000 medical content and item type feature constraints for an examination of about 300 items. In these types of situations involving high-stakes, content-critical examinations, it is simply not feasible to break down the test specifications at the module level. Test-level specifications are needed to ensure that the combinations of modules achieve the desired balance of content and many other relevant features.

<u>Both</u> issues can be resolved by creatively using the CAST *pathways* as surrogate "test forms". For example, Figure 1 provided one easy form (A+B+E+H), one moderate difficulty form (A+C+F+I), and one hard form (A+D+G+J). That is, we can simply ignore the secondary pathways and design three simultaneous test forms, each a

different level of average difficulty, and where each shares a common block (module or testlet) of items.

Designing Target Test Information Functions for CAST Panels

In this section, I present a general overview of some simple strategies for creating target test information functions for the various *primary pathways* in a CAST panel. Realize that using separate statistical targets for each module is also an alternative that Luecht and Nungester (1998) discuss in the context of a "bottom up" test assembly strategy.

The literature on using IRT test information functions (TIFs) for automated test assembly is replete with examples (van der Linden, 1987, 1994; van der Linden and Boekkooi-Timminga, 1989; Adema, 1990; Luecht, 1992; Luecht and Hirsch, 1992; Luecht, 1998; Armstrong, Jones, Li, and Wu, 1996).

Assume that a particular IRT model, such as the three-parameter (3P) model,

$$P\left(u_i \middle| \theta, a_i, b_i, c_i\right) \equiv P_i = c_i + \frac{1 - c_i}{1 + \exp\left[-Da_i\left(\theta - b_i\right)\right]} \qquad (1)$$

fits the data (Lord, 1980). In Equation 1, the usual item parameters are denoted as $a_i$, $b_i$, and $c_i$, with individual items indexed by $i$. $\theta$ is the latent proficiency trait and $D$ is a scaling constant ($D = 1.0$ for a logistic response function or $D = 1.7$ to approximate a normal ogive response function).

As Birnbaum (1968a, 1968b) demonstrated, when θ is estimated by maximum likelihood from dichotomously scored item responses, $u_i$, the item information function for the 3PL model is

$$I\left(\theta_j ; u_i\right) \equiv I_i(\theta) = \frac{D^2 a_i^2 Q_i \left(P_i - c_i\right)^2}{P_i \left(1 - c_i\right)^2} \tag{2}$$

noting that $Q_i \equiv 1 - P_i$. For the 1PL and 2PL models, we can make the obvious simplifications to the information function (see Hambleton and Swaminathan, 1985).

The item information functions can be summed to produce a test information function (TIF). The reciprocal of the TIF is the error variance of the estimated θ score; i.e.,

$$\text{TIF} = \sum I_i(\theta) = \frac{1}{\text{var}\left(\hat{\theta} \middle| \theta\right)} \tag{3}$$

Therefore, by targeting the TIF value we want at various regions of the proficiency scale (θ), we effectively control the amount of error variance (precision or lack therepof) of the estimated scores.

*Here, CAST differs from computer-adaptive testing in a very fundamental way.* Under traditional CAT, the implicit "target" TIF is the maximum information possible at the final estimate of proficiency, so items are sequentially selected to maximize Equation 3 with respect to the provisional estimates of the proficiency. The process of maximizing information in CAT requires a heuristic due to the use of provisional proficiency estimates.

Under CAST, we explicitly choose one or more ***target*** TIFs that will provide consistent score precision over time, rather than the maximum possible information in an item bank. We then use automated test assembly (ATA) item selection procedures to build each panel in order to achieve our target test information function(s). So, rather than maximizing information, CAST panels designs are more apt to use robust, average test information targets so that parallel score precision can be maintained over time and across panels (Luecht, 1992, 1998; Luecht & Nungester, 1998).

Any number of sound strategies can be used to generate target TIFs for a CAST panel (e.g., constrained bootstrapping of the item bank or simulating an adaptive test for a limited number of $\theta$ values). With CAST, the key to generating targets is to focus on the *primary pathways* within the panel. For example, the panel configuration shown in Figure 1 has four stages, but only three primary pathways (shown by the solid connector lines): (1) Pathway A+B+E+H, the easy pathway; (2) Pathway A+C+F+I, the moderate difficulty pathway; and (3) Pathway A+D+G+J, the hard pathway. Figure 2 shows what the target TIFs for the three primary pathways might look like.

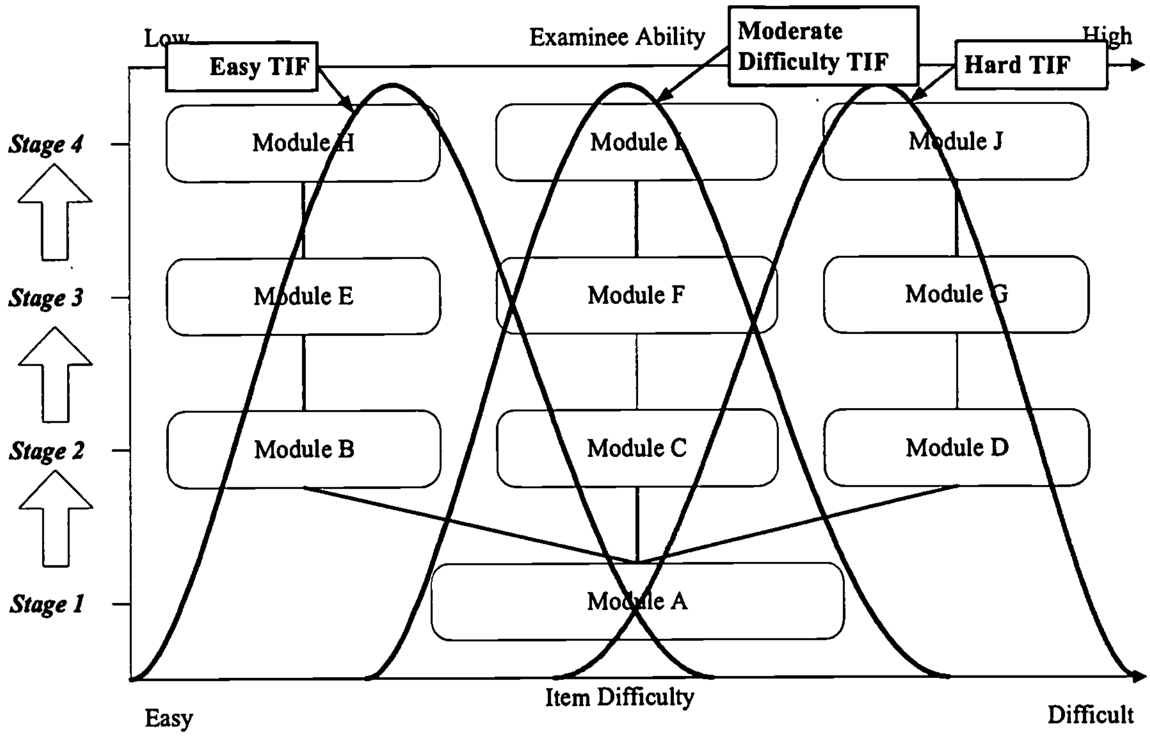


Figure 2. Possible Test-Level Target TIFs for the Ten-Module, Four-Stage CAST Panel

Conceptually, it may be useful to think about each of the primary pathways as a separate test form; in our example, there is one easy form, one moderate difficulty form, and one hard test form. Although some examinees may indeed transition along the secondary pathways–i.e. be routed along the dashed lines in Figure 1–most "well-behaved" examinees[3] will follow one of the three primary pathways, from a probabilistic perspective. We therefore need to generate a separate target TIF for each

[3]Examinees who do not exhibit consistent patterns of performance can be flagged as "aberrent" (i.e. model-based misfitters). There are numerous plausible reasons for misfit, ranging from a misspecified IRT model to cheating or illness on a particular test section. By precluding examinees from moving more than adjacently, between stages, extreme cases of misfit can be curtailed.

pathway. Below, I present three strategies for creating the test information functions for pathways.

Each of the three strategies can be implemented by using a technique described in Luecht (1992) for generating target TIFs at specific locations on a score scale. For convenience, I will call this the Average Maximum Information (AMI) technique. The AMI technique is conceptually similar to simulating multiple adaptive tests, without replacement. The method is as follows.

1. Locate a particular point on the $\theta$ scale. This point will generally correspond to the location of desired maximum information (modal value or peak) of a given TIF. For example, if I wanted to have my "moderate" target TIF have maximal information at the mean of the ability distribution–assuming a normal (0,1) distribution–I would choose a value of $\theta_M = 0.0$. Further suppose that we wanted the easy pathway to have a maximum information at the 30th percentile with a corresponding location of $\theta_E = -0.52$ and the hard pathway to have maximum information at $\theta_H = +0.84$ (the 80th percentile).
2. For each item in the item bank, compute the value of the information function (Equation 2) at one of the selected locations where we want maximum information–for example, at $\theta_j$, where $j \in \{E, M, H\}$. This can be done with customized software, in a database package with computed fields, or with a spreadsheet.

3. Sort the item bank in descending order by the computed item information value, focusing on only one of the selected locations (e.g., $\theta_M$). Multiple information computations and data sorts will be needed to obtain the maximally informative items at each of the other $\theta$ locations.

4. Given a particular quantity of items corresponding to either the test length or some smaller number, denoted here as $n$ (e.g., $n$ could also be the size of a single module or combination of several modules), choose some number of maximally informative replications, without replacement, which we can denote as $m$. Depending upon the size of the item bank, $m \geq 5$ is a reasonable minimum requirement. In general, the larger the value of $m$, the more robust will be the derived information target (Luecht, 1992). For example, if $n = 20$ items and we elect to create $m$=10 replications of the simulated test, without replication, we would choose the $n \times m = (20)(10) = 100$ most informative items at the selected $\theta$ location. [Author's note. This procedure mimics using an adaptive item selection algorithm to build $m$ non-overlapping test forms of length $n$ that are maximally informative at a particular value $\theta$.]

5. Compute the sum of the test information at each of several selected ability points, $\theta_k$, $k$=1,...,$K$ and divide by $m$ to obtain the mean target TIF. That is, compute $\mathrm{TIF}_{jk}=T_j(\theta_k)$, where

$$T_j(\theta_k) = \frac{\sum_{i=1}^{n \cdot m} I_i(\theta_k)}{m} \tag{4}$$

In most cases, a grid of $10 \leq K \leq 20$ equally spaced points from from $-3.0$ to $+3.0$ will suffice to adequately represent the target TIF across a reasonable range of $\theta$.

Once computed, the TIFs can be summed with other TIFs or subdivided as needed to generate the final targets for each pathway. Keeping this general approach in mind, I can now describe three strategies for generating TIFs in the CAST context.

<u>The Middle-Out (MO) Strategy</u>

This strategy is useful when there a panel has an odd number of primary pathways (e.g., Figure 1). The strategy steps can be enumerated as follows.

1. Generate an average target test information function for the center-most primary pathway (for example, A+C+F+I in Figure 1), using the AMI method described above. The initial location value of $\theta$ to use is $\theta_M$. The test length, $n$, should be the sum of the item counts for all of the modules in the moderate difficulty pathway.

2. For equal-sized modules at each stage, divide the computed target $\mathrm{TIF}_{Mk}=T_M(\theta_k)$ values by the number of stages for the $k$=1,...,$K$ grid of points on the proficiency scale. For unequal-sized modules per stage, multiply the $\mathrm{TIF}_{Mk}$ values at each grid point by the proportion of the test length corresponding to the size of the Stage 1 module (e.g., the proportion of items in Module A). This is the Stage 1 $\mathrm{TIF}_{M(1)k}$. Retain both the full-length test information function for the middle pathway, $\mathrm{TIF}_{Mk}$ and the information function for only Stage 1, $\mathrm{TIF}_{M(1)k}$

3. Repeat the AMI procedures for the remaining "outer" locations, selecting the maximally informative items at $\theta_E$ and $\theta_H$, however, proportionally reduce the test length, *n*, by the size of the Stage 1 module (Module A). The same items can be reused, if needed, for each of the "outer" pathways. This will produce two addition test information functions, $TIF'_{Ek}$ and $TIF'_{Hk}$, where each is based on the test length, less the size of the Stage 1 module.

4. Add the computed "outer" pathways test information functions to the Stage 1 $TIF_{M(1)k}$ to obtain the full-length test target functions–that is, $TIF_{Ek}=TIF'_{Ek} + TIF_{M(1)}$ for the *easy* pathway and $TIF_{Hk}=TIF'_{Hk} + TIF_{M(1)k}$ for the *hard* pathway. The $TIF_{Mk}$ for the middle pathway is the full-length target test information function for the moderate difficulty pathway.

The Common-First-Module (CFM) Strategy

This strategy is similar to the MO strategy, but allows somewhat more control over the Stage 1 target TIF. It can be used with even or odd numbers of pathways and when the nature of the test information targeting is skewed or uncentered in some fashion (e.g., if we had four pathways or if we had three pathways with Module A presented as an "easy" module).

1. Determine the size of the Stage 1 module. Use the AMI method to determine the target, $TIF_{M(1)k}$, for this Stage 1 module. Filter out the items in the item bank used to generate this target test information function, $TIF_{M(1)k}$, $k$=1,...,$K$.

2. Proportionally reduce the total test length by the size of the Stage 1 module. Using the remaining items in the item bank, repeat the AMI procedure for all of the locations–i.e., for $\theta_E$, $\theta_M$, and $\theta_H$, using the proportionally reduced test length, $n$-$n_1$. The corresponding, reduced-length targets can be denoted as $TIF'_{Ek}$, $TIF'_{Mk}$, and $TIF'_{Hk}$.

3. Add the Stage 1 test information function to each of the reduced test-length information functions to obtain the three full-length target test information functions for the various pathways. That is $TIF_{Ek}=TIF'_{Ek}+TIF_{M(1)k}$ for the easy pathway, $TIF_{Mk}=TIF'_{Mk}+TIF_{M(1)k}$ for the moderate difficulty pathway, and $TIF_{Hk}=TIF'_{Hk}+TIF_{M(1)k}$ for the hard pathway.

The Separate-and-Average-the-First (SAATF) Strategy

This method is probably the most straight-forward method to implement, but provides the least degree of control over the design of the target TIFs.

1. Use the AMI method to produce TIFs for each of the primary pathway $\theta$-locations, $\theta_E$, $\theta_M$, and $\theta_H$, at the full-test length. That is, compute $TIF_{Ek}$, $TIF_{Mk}$, and $TIF_{Hk}$, at $k$=1,...,$K$ grid points.

2. For equal-sized modules at each stage, divide the target $TIF_{jk}$ values by the number of stages at each of the grid points ($\theta_k$, at $k$=1,...,$K$ points on the proficiency scale). For unequal-sized modules per stage, multiply each of the $TIF_{jk}$ values by the proportion of the test length corresponding to the size of the Stage 1 module (e.g., the proportion of items in Module A). In our example

(Figure 1) this would produce three separate Stage 1 test information functions, $TIF_{E(1)k}$, $TIF_{M(1)k}$, and $TIF_{H(1)k}$. Average the three (possibly disparate) TIF values at each of the grid points to produce a single target $TIF_{(1)k}=\sum TIF_{j(1)k}/3$ for Stage 1.

3. Proportionally multiply each total-test-length TIF by percentage of items remaining, $p'=(n-n_1)/n$ in Stages 2, 3 and 4. That is, $TIF'_{jk}=p'(TIF_{jk})$.

4. Add the Stage 1 $TIF_{j(1)k}$ values to the various reduced test-length test information values to obtain the full-length target TIFs for the various pathways. That is $TIF_{Ek}=TIF'_{Ek}+TIF_{(1)k}$ for the easy pathway, $TIF_{Mk}=TIF'_{Mk}+TIF_{(1)k}$ for the moderate difficulty pathway, and $TIF_{Hk}=TIF'_{Hk}+TIF_{(1)k}$ for the hard pathway.

Creating Module-Level TIFs

The MO, CFM, and SAATF strategies can also be used to generate module-level test information function (TIF) targets. First, generate the full-length target TIFs for each primary pathway, as described in the previous section. Then, divide each target TIF by the proportional size of the various stages. Because the IRT test information functions are additive, we can break the larger TIFs apart as easily as we put them together.

Categorical and Other Test-Level Constraints

Most well-designed tests have documented specifications for the measurement properties and statistical characteristics of the test as well as other content features and attributes which test developers consider important in building new test forms. None

of that changes as a function of moving to CAST or any other type of computer-based test.

Categorical attributes of items or sets are taxonomically-coded features stored in the item bank (e.g. content codes, cognitive level codes, item type codes, or item author identification codes). These features are typically controlled at the test level by introducing "constraints" as part of the test specifications used in automated test assembly. For example, the required frequencies or proportions of items for a test form covering various subject areas typically taught in a high school mathematics course could be stated as either constraints on the range of items to include in each subject area or as exact frequencies (e.g. 10 to 15 items in intermediate algebra or exactly 12 items items in geometry). For some tests, the content specifications may be include only a few general categories. In other cases, the content specifications may cover a very long content outline with numerous levels for each category and many auxiliary classification taxonomies having additional constraints (e.g. item types or formats, cognitive levels, and item authors). In CAST, categorical constraints can be introduced at the test level, at the module level, or both.

Perhaps the most common constraint in test assembly is the test length. The test length (or module length) can be constrained to equal a fixed value (e.g., exactly 100 items) or a variable number where the latter could be specified as a minimum and maximum test length constraints (for example, at least 90 items but no more than 120 items).

Other non-categorical constraints are quantities such as word counts and average time per item or case. These quantities can be computed for each item and constrained as sums over items at the test level. It is common to specify constraints that indicate acceptable ranges of these quantities (e.g. 2500 to 2800 words).

Two additional types of special constraints encountered in test assembly are: (1) limits on the item reuse frequencies across test forms (or across modules, pathways, stages, and panels); and (2) item exclusions. Constraining the item reuse frequency is important in that it controls the exposure of test materials to examinees. If items are used too often and on too many test forms, examinees may conspire to cheat by memorizing and sharing the items. Most high stakes testing programs have to pay very serious attention to the issue of item reuse. I will discuss these reuse constraints more in the next two sections.

Item exclusions are usually rules relating to the relationships among the items. For example, an "all-or-none" exclusionary rule can be established so that any item set selected must be taken as is (i.e. with all its associated items) or not at all. The next section briefly addresses these types of exclusion.

Another set of exclusionary rules may govern the use of item "enemies" alluded to above. For example, two items which clearly cue one another probably should not appear on the same test form. The exclusionary rule would make such pairs or clusters of "enemies" mutually exclusive on the same test form. By assigning "enemies" a

common attribute, constraints can be placed on reuse (e.g., a maximum of only one item can be selected from a given "enemy set", see Luecht, 1998).

Under CAST, we need to establish the constraints for all of the quantitative and categorical attributes for a full-length test form. There may be a few constraints, hundreds of constraints (Luecht, Nungester & Hadadi, 1996; Luecht & Nungester, 1998) or even thousands of constraints (Hadadi, Swanson & Luecht, 1999). Each pathway inherits the common set of constraints, however, there are different statistical target TIFs for each pathway. I will say more about this in the next section on automated test assembly.

### Using Automated Test Assembly to Build CAST Panels

Having a sophisticated test design technology is virtually useless without a means to feasibly generate test forms from an item bank. Although automated test assembly (ATA) has been effectively used in small-scale contexts to generate a limited number of fixed-length test forms, there have been very few successful demonstrations of applications of ATA technologies to CAST and multi-stage testing problems–exceptions being the work by Luecht et al at the National Board of Medical Examiners and some of the work by Stocking et al at ETS, involving CAT and CMT. Understand that CAST dramatically changes nature and scope of the test assembly process by requiring ATA to mass produce panels.

The ATA problems inherent in CAST and other forms of multi-stage testing can become massive, especially with thousands of items in an item bank, hundreds or thousands of constraints (i.e., categorical test specifications), multiple targets, and the need to replicate the panels many times over.

In this section, I describe a heuristic known as the normalized weighted absolute deviations heuristic (NWADH) and describe how it can be used to build CAST panels. The NWADH is one of several "successful" heuristics for large-scale test assembly; another being the weighted deviations model presented by Swanson and Stocking (1993).

The NWAD heuristic has been implemented in a proprietary, large scale test assembly package used by the National Board of Medical Examiners and in a number of other test assembly packages written for PCs by the author. The heuristic has also been implemented for the 3PL model in a computer program called CASTISEL, a DOS-based shareware program (Luecht, 1996; 1999). Various versions of CASTISEL have been successfully used to build multi-stage test forms for a number of examination programs and research projects, including building CAST field test forms for the United States Medical Licensing Examination Step 1 in 1997 (Federation of State Medical Boards and National Board of Medical Examiners).

<u>The NWAD Heuristic</u>

Given the previous introduction to "targets" and "constraints", the general optimization problem solved by the NWADH can be outlined as follows. Let $u_{ik}$ denote

an item information function (or any other relevant attribute) for $i$=1,...,$I$ items in an item data base, evaluated at $k$=1,...,$K$ points. For a particular test form comprised of $n \leq I$ items, the corresponding test attribute can be expressed as

$$\sum_{i=1}^{n} u_{ik},\ k=1,\dots,K$$

where the attributes, $u_{ik}$, i=1,...n, are assumed to be algebraically additive for all items in the test (see Equation 3). A corresponding target test function is defined as $T_k$. That is, $T_k$ denotes a corresponding test information function that we would like to meet (e.g. a test information function to be matched for building parallel test forms over time). An objective function could now be defined to

$$\text{minimize} \sum_{k=1}^{K} \left| T_k - \sum_{i=1}^{I} x_i u_{ik} \right| \tag{5}$$

subject to the two simple constraints:

$$\sum_{i=1}^{I} x_i = n; \tag{6}$$

$$x_i \in \{0,1\},\ i=1,\dots,I. \tag{7}$$

In Equation 7, the $x_i$, are decision variables for selecting the n items; this constraint on the test length is stated explicitly in Equation 6. That is, $x_i = 1$ if the item is to be included in the test, otherwise, $x_i = 0$.

To formally implement the NWADH algorithm, we need to change the absolute deviation minimization problem in Equation 5 to a maximization problem and introduce some additional notation. The item selection process can be managed at the unit level, where j=1,...,n objective functions are to be maximized. That is, for a series of n optimization models,

$$\text{maximize} \sum_{i=1}^{I} e_i x_i \tag{8}$$

subject to

$$\sum_{i=1}^{I} x_i = j \tag{9}$$

$$x_{i_1} = x_{i_2} = \quad = x_{i_{j-1}} = 1 \tag{10}$$

$$x_i \in \{0,1\},\ i=1,\ldots,I. \tag{11}$$

where, $e_i$ is a variable coefficient,

$$e_i = 1 - \frac{d_i}{\sum_{i \in R_{j-1}} d_i}\ ,\ i \in R_{j-1} \tag{12}$$

and

$$d_i = \sum_{k=1}^{K} \left| \left( \frac{T - \sum_{r=1}^{I} u_{rk} x_r}{n - j + 1} \right) - u_{ik} \right| \quad ; i \in R_{j-1}. \tag{13}$$

In Equations 12 and 13, $R_{j-1}$ is defined as a set of indices for the remaining items in the item bank after excluding the selected $j - 1$ items.

The localized optimization model must be solved at each item selection, $j$=1,...,n, since Equations 8 to 13 only relate to the selection of the current item, $j$. As each new item is selected, Equation 8 is incremented via Equations 12 and 13. Finally, the expression in Equation 13,

$$\frac{T_k - \sum_{r=1}^{I} u_{rk} x_r}{n - j + 1} ,$$

provides the current value of the target function, after removing previously selected items (evaluated, as before, at $k$=1,...,$K$ points).

Finally, we normalize the coefficients, as shown in Equation 12, by dividing the $d_i$ variables by their sum over all eligible item. The normalization transforms the absolute difference function into a *proportional* quantity. This simple transformation allows the NWADH to be easily extended to deal with any number and type of content or other categorical attributes and can also deal with multiple content dimensions or

facets and levels within those dimensions (e.g. content outline sublevels). For purposes of convenience and clarity, only a single content dimension is considered here.

Let G denote the total number of content categories for a particular content or other taxonomical dimension with the individual categories indexed $g = 1,\ldots,G$. Let $v_{ig} \in \{0,1\}$ denote the binary incidence of an item having a particular categorical content attribute, $g=1,\ldots G$, for all items, $i=1,\ldots,I$ in the item bank. That is, $v_{ig}$ equals one if the item belongs in the category or zero if not. Finally, let $Z_g^{[min]}$ represent some minimum constraint quantity and $Z_g^{[max]}$ represent some maximum constraint quantity for each of the $g=1,\ldots,G$ content categories. For any particular categorical content attribute, the sum,

$$\sum_{i=1}^{I} v_{ig},\ g=1,\ldots,G,$$

provides the availability of items in the item having that attribute. Note that it is assumed–rather logically–that the availability of items is greater than zero for all specified categories and that if

$$\sum_{i=1}^{I} v_{ig} < Z_g^{[min]}, \tag{14}$$

the constraint, $Z_g^{[min]}$ will be adjusted so that

$$\sum_{i=1}^{I} v_{ig} \geq Z_g^{[min]}. \tag{15}$$

Recall that $x_i$ was previously defined as a binary decision variable denoting whether each item, i=1,...,I, was selected or not. The count of items selected for each content category in the NWADH sequence (i.e. up to the preceding item selection, j - 1) can therefore be computed as

$$\sum_{i \in R_{j-1}} v_{ig},\ g=1,\ldots,G.$$

This sum can be used to empirically determine a set of weights, $W_g$, g=1,...,G. These category weights can take on either user assigned (e.g. integer weights or points) or empirically determined values (e.g. proportions based on remaining availabilities of items in the bank after each item selection). A simple, but effective weighting scheme is given as follows. Assume that $Z_g^{[min]} < Z_g^{[max]}$. Using a single item point assignment scheme, the weights for each category, $W_g$, could be assigned to take on one of three values:

(i) if $\sum_{i \in R_{j-1}} v_i \geq Z_g^{[max]}$ then $W_g = 0$;

(ii) if $Z_g^{[min]} \leq \sum_{i \in R_{j-1}} v_i < Z_g^{[max]}$ then $W_g = 1$; or,

(iii) if $\sum_{i \in R_{j-1}} v_i < Z_g^{[min]}$ then $W_g = 2$ (g=1,...,G).

At each iteration of the NWADH, the weights are accumulated by each unselected item. Therefore, items belonging to the categories that have not met the minimum constraint, $Z_g^{[min]}$, get more weight than those that have met the minimum but which do not exceed the maximum constraint, $Z_g^{[max]}$, g=1,...,G. Items in categories that are at or in excess of the maximums get no weight, whatsoever.

It may seem strange that there are no negative or "penalty" weights assigned for exceeding any maximum, $Z_g^{[max]}$. Instead of penalizing items which violate particular upper bound constaints, an alternative approach was devised which works quite well in practice. That approach is to reward all the items which do not have the categorical attribute which is at or in excees of $Z_g^{[max]}$.

Let $W^{[max]}$ represent the maximum value of the weights across all G categories. An approximate a complement to $W_g$, denoted $\underline{W}_g$, can be computed as

$$\underline{W}_g = W^{[max]} - \frac{1}{G}\sum_{i=1}^{G} W_g . \tag{16}$$

As the constraints in particular category are met, the right-most average weight term approaches $W^{[max]}$; correspondingly, $\underline{W}_g$ approaches zero. Items not belonging to any of the specified (i.e. constrained) categories are rewarded with what amounts to "bonus" points for not contributing to categories at or in excess of the maximums. Therefore, instead of penalizing items for violating upper bound constraints, the

complement weight proportionally rewards all the other items which do not belong to the violated category or categories.

Now, let $c_i$ be the accumulated content weights for each unselected item in $R_{j-1}$. The weights, $W_g$ and $\underline{W}_g$, are used to compute $c_i$ as follows:

$$c_i = v_{ig} W_g + (1 - v_{ig}) \underline{W}_g ; i \in R_{j-1}, g=1,\ldots,G. \tag{17}$$

This new item-level variable can be normalized for all unselected items (i.e. items remaining in the set of unselected items, $R_{j-1}$). The normalized variables can then be used in conjunction with the normalized statistical coefficient given in Equation 12 to define to new variable coefficient to be maximized in the objective function. That is,

$$e_i^* = \left(1 - \frac{d_i}{\sum_{i \in R_{j-1}} d_i}\right) + \frac{c_i}{\sum_{i \in R_{j-1}} c_i} \qquad , i \in R_{j-1}, \tag{18}$$

where $e_i^*$ can now be substituted for $e_i$ into Equation 8. For some applications, user-assigned, proportional weight coefficients can be incorporated into the composite function in Equation 18 to reflect the importance of the meeting statistical versus categorical or content specifications. By adding new terms to Equation 18 to accomodate multiple categorical or quantitative targets or constraints, the NWADH can be extended to handle some very large test assembly problems.

Where content or other categorical attributes are fixed as *primary* test construction requirements having exact quantities along each pathway, it is possible to

significantly speed up the heuristic by implementing prioritized searches for the items within particular categories. For example, suppose that $Z_g$ (defined earlier as a constraint miminum or maximum value) is now treated as a fixed quantity so that every test form must have exactly $Z_g$ items in the g=1,...,G categories.

A "need-to-availability" ratio for each category can be computed as

$$A_{(j-1),g} = \frac{Z_g - \sum_{i \in R_{j-1}} v_{ig}}{\sum_{k=1}^{I} v_{kg} - \sum_{i \in R_{j-1}} v_{ig}}, \quad g=1,\ldots,G,\ j=1,\ldots,n. \tag{19}$$

If the denominator of Equation 19 is zero, there are no more items remaining in the category and $A_{(j-1),g}$ should be set to zero. This "need-to-availability" ratio can updated after each item selection where large values indicate higher priority than smaller values. At each iteration, the category with the maximum value of $A_{(j-1),g}$ (i.e. the greatest need-to-availability) is searched and the NWADH is only applied to items in that category. This approach can significantly improve the speed of the overall solution since the more computationally intensive NWADH only has to be applied to a small subset of items each time.

When prioritized in this manner, items belonging to categories that have the greatest need and smallest availability tend to be chosen earlier than those categories having low demand or a large surplus of items on hand. Where the demand is high and

the supply is small (e.g. the specifications call for 5 items in a subject area and there are only 5 such items in the bank), there is little choice about selecting the items. The real question is not "if", but "when" those items be chosen. This prioritization mechanism forces those high priority items into the solution early, allowing the NWADH to have more flexibility further on to build around them.

The NWADH can handle the concurrent item selections for CAST (i.e., selecting items along multiple pathways and for multiple panels) by implementing separate objective functions for each pathway and for each replication of the pathways over panels. It is also possible to allow items to appear within multiple pathways for the same panel. An upper bound can be placed on the number of reuses allowed per item (or globally for all items in the item bank). This upper bound usage constraint can even be converted into a proportion of maximum allowable use and incorporated into the variable coefficient term so that items having no or smaller amounts of reuse across test forms are more likely to be chosen for a particular form, all other considerations being equal.

Item sets are multi-item units (e.g. several items associated with a reading passage, vignette, or other common stimulus). From the perspective of how the NWADH functions, dealing with item sets is an almost trivial generalization. The objective function for the heuristic can be modified to locally optimize the selection of multiple items as easily as a single item. In fact, item sets may carry their own class-

level categorical attributes (e.g. type of reading passage) which can also be entered as constraints.

To summarize, in order to build CAST panels, it is virtually essential to use ATA software. Programs like CASTISEL, that employ optimization algorithms or heuristics like the NWADH, need to ideally have four capabilities in order to build CAST panels. First, the software should be able to *simultaneously* optimize more than one objective function in the presence of multiple and different target information functions (i.e. multiple target functions within panels and replications of those targets across panels). Second, the software may possibly need to meet different content specifications and constraint systems for various modules or pathways (i.e., not be limited to module-level OR test-level pathway constraints and specifications). Included should be ways to deal with item sets and with "enemies", at least as specially classified and constrained attributes. Third, the software should have few practical limitations on the number of categorical dimensions or constraints used in a given problem run. Simultaneously managing several thousand constraints and *huge* item banks ought to be feasible on a PC, with reasonable execution times. Fourth, the ATA software must be capable of building many replications (e.g. perhaps as many as 100 different versions) of the modules, pathways, and panels with item overlap carefully controlled within and across panels.

This listing of capabilities is more of a "wish list" than a reality. CASTISEL does not provide all of these capabilities, and, to my knowledge no commercially available or

shareware software can handle the complete scope of these capabilities either. However, CASTISEL does provide many of these capabilities and is freely distributed by the author.

## Item Exposure Control and Randomized Assignment of Panels

The issue of item exposure in CAT has become one of the central research topics in measurement. Examinees can and do cheat, including conspiring to memorize and share items from otherwise "secure" item banks. In CAT, item exposure controls (e.g., Revuelta and Ponsoda, 1998) more-or-less serve as "penalty functions" to buffer the effect of selecting items that maximize the test information function–i.e., probabilistically or otherwise constrain the reuse or exposure of the items, while the item bank is active.

CAST has three distinct advantages in this regard. First, by using robust, average test information targets for the modules or pathways, CAST panel designs can naturally buffer the exposure of items, since the most informative items will be more uniformly distributed over panels. Second, by explicitly including ATA constraints on item reuse within panels (across pathways) and across panels, we can directly achieve control that over the amount of item exposure, item-by-item. Third, we can empirically determine our exposure risks, because we pre-construct the CAST panels. For example, if we limit using a particular item to 30 times per 100 panels, it has a maximum exposure rate of 0.30, assuming uniform random assignment of the panels.

We can likewise compute conditional exposures by constraining the reuse across pathways within panels or even across stages. It is even possible to directly incorporate conditional exposure controls into the NWADH (Luecht, 1998) as part of the ATA process.

However, the real advantage of CAST lies in pre-constructing the panels. Because we know, beforehand, which panels will be active, we can compute the real exposure risks across and within those panels. That is, we include checks for exposure risks as part of the normal quality assurance process. Items with excessive reuse (exposure risk) can be scrutinized and appropriate substitutions made, before the panels are activated.

Another simple but highly effective administrative capability in CAST is random assignment of the panels to examinees. Because the CAST panels are legitimate test administration units (at least from an administrative database perspective) they can be assigned "form numbers" and randomly assigned to particular examinees. This is a trivial capability of CAST with important implications. Even repeat testers can be convenient handled by limiting the "panel pool" to panels having minimal overlap with previous pathways the examinees may have seen. The simplicity of CAST to address these rather complex security issues is one of its most appealing aspects.

## Data Management and Control

Computer-adaptive testing and computer-mastery testing can put enormous stress on an organization's examination processing systems, especially if those systems were designed to handle a limited number of paper-and-pencil test forms each year. The steady flow of examinee registration data to-and-from data centers and test sites and the all-but-random stream of examinee data returning for processing can create huge bottle-necks and tedious data management to ensure the quality and integrity of all of the data (confirming complete examinee response records, verifying answer keys and scoring, etc.).

Even with powerful computer systems and sophisticated database software, serious data management problems can and do surface. Unreconciled data, missing and partial examinee records, lack of control over repeat test-takers, mismanagement or failure to catch miskeyed item data, rescoring hassles, legal challenges requiring total reproduction of the examinees testing sequence and response patterns, justifying the poor quality of test forms to test committees or constituencies, or claims of "unfairness" when tests are generated by randomization algorithms or computer-adaptive item selection algorithms, are just some of the problems that do occur–sometimes more as routine situations than as exceptions. CAST cannot solve all of these problems.

CAST implements a highly controllable set of data structures: the panels and modules within those panels, with pathways explicit to the panel. That degree of control is desirable on many technical levels of data management. From the many

advantages of assigning unique "form" identifiers to panels, to the capabilities to evaluate scoring and answer keys by using simulated response patterns to "master test" the primary pathways in all active panels, CAST is about systematic control of the test design and administration process. CAST is also thoroughly consistent with modern object-oriented database management perspectives.

## Discussion

CAST is not a panacea for computer-based testing and is certainly not the optimal multi-stage test design for every testing program. It is a straight-forward test development framework for mass producing and administering structured multi-stage computer-adaptive and computer-mastery tests where quality assurance and security can be checked before the tests are administered. It also offers some subtle advantages in terms of security and data management.

One desperate need is ATA software. Without capable ATA software, multi-stage techniques like CAST are nice concepts with limited utility. Shareware computer programs like CASTISEL are useful for demonstrating CAST and may even work for small-scale research projects. However, we need the applied apabilities described earlier (multiple objective functions and the capability to handle thousands of items and constraints).

As CAST evolves and grows in use, its merits and faults will become more apparent. For now, it seems to be a reasonable idea.

## References


Adema, J. J. (1990). *Models and Algorithms for the Construction of Achievement Tests.* Doctoral Dissertation. Enschede, The Netherlands: FEBO.

Armstrong, R. D., Jones, D. H., Li, X., and Wu, I-L. (1996). A study of a network-flow algorithm and a non-correcting algorithm for test assembly. *Applied Psychological Measurement, 20,* 89-98.

Birnbaum, A. (1968). Test scores, sufficient statistics, and the information structures of tests. In F. M. Lord and M. Novick (Eds.). *Statistical Theories of Mental Test Scores.* Reading, MA: Addison-Wesley.

Birnbaum, A. (1968). Estimation of an ability. In F.M. Lord and M. R.Novick (Eds.). *Statistical Theories of Mental Test Scores,* pp. 453-479. Reading, MA: Addison-Wesley Publishing.

Hambleton, R. K. & Swaminathan, H. (1985). *Item Response Theory: Principles and Applications.* Boston: Kluwer.

Hadadi, A., Swanson, D. B., Luecht, R. M. (April, 1998). *Practical issues in using automated test assembly (ATA) for large-scale testing*. Paper presented at the Annual Meeting of the National Council on Measurement in Education, Montreal, Quebec.

Lord, F. M. (1980). *Applications of Item Response Theory to Practical Testing Problems*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Luecht, R. M. (1992, April). *Generating target information functions and item specifications in test design*. Paper presented at the Annual Meeting of the National Council on Measurement in Education, San Francisco, CA.

Luecht, R. M. (1998) Computer-assisted test assembly using optimization heuristics. *Applied Psychological Measurement, 22*, 224-236.

Luecht, R. M. & Hirsch, T. M. (1992). Item selection using an average growth approximation of target test information function. *Applied Psychological Measurement., 16*, 41-51.

Luecht, R. M. & Nungester, R. J. (1998). Some practical applications of computer-adaptive sequential testing. *Journal of Educational Measurement, 35*, 229-249.

Luecht, R. M., Nungester, R. J. & Hadadi, A. (1996, April). *Heuristic-Based CAT: Balancing Item Information, Content and Exposure*. Paper presented at the Annual Meeting of the National Council of Measurement in Education, New York, NY.

Luecht, R. M., Nungester, R. J. Swanson, D. B. & Hadadi, A. (April, 998). *Examinee Performance, Perceptions, and Pacing on a Computer-Adaptive Sequential Test*. Paper presented at the Annual Meeting of the National Council of Measurement in Education, San Diego, CA.

Reveulta, J. & Ponsoda, V. (1998). A comparison of item exposure control methods in computerized adaptive testing. *Journal of Educational Measurement, 35*, 311-327.

Stocking, M. L. & Swanson, L. (1993). A method for severely constrained item selection in adaptive testing. *Applied Psychological Measurement, 17*, 277-292.

Stocking, M. L., Swanson, L. & Pearlman, M. (1993). Application of an automated item selection method to real data.. *Applied Psychological Measurement, 17*, 167-176.

Swanson, L. & Stocking, M. L. (1993). A model and heuristic for solving very large item selection problems. *Applied Psychological Measurement, 17*, 151-166.

van der Linden, W. J. (1987). Automated test assembly using minimax programming. In W. J. van der Linden (Ed). *Research Report 87-2*. Enschede, The Netherlands: University of Twente.

van der Linden, W. J. (1994). Computerized educational measurement. In T. Husen and T. N. Postlethwaite (Eds.). *International Encyclopedia of Education (2nd Ed.)* (pp. 992-998). Oxford: Elsevier.

van der Linden, W. J. & Boekkooi-Timminga, E. (1989). A minimax model for test design with practical constraints. *Psychometrika, 54*, 237-247.

Wainer, H. & Kiely, G. L. (1987). Item clusters and computerized adaptive testing: a case for testlets. *Journal of Educational Measurement, 24(3)*, 185-201.

Wald, A. (1947) *Sequential Analysis*. New York: Wiley

**U.S. Department of Education**
*Office of Educational Research and Improvement (OERI)*
*National Library of Education (NLE)*
*Educational Resources Information Center (ERIC)*

ERIC®

TM031243

# REPRODUCTION RELEASE

(Specific Document)

## I. DOCUMENT IDENTIFICATION:

| | |
|---|---|
| Title: Implementing the Computer-Adaptive Sequential Testing (CAST) Framework to Mass Produce High Quality Computer-Adaptive and Mastery Tests | |
| Author(s): Richard M. Luecht | |
| Corporate Source: n/a | Publication Date: n/a |

## II. REPRODUCTION RELEASE:

In order to disseminate as widely as possible timely and significant materials of interest to the educational community, documents announced in the monthly abstract journal of the ERIC system, *Resources in Education* (RIE), are usually made available to users in microfiche, reproduced paper copy, and electronic media, and sold through the ERIC Document Reproduction Service (EDRS). Credit is given to the source of each document, and, if reproduction release is granted, one of the following notices is affixed to the document.

If permission is granted to reproduce and disseminate the identified document, please CHECK ONE of the following three options and sign at the bottom of the page.

| The sample sticker shown below will be affixed to all Level 1 documents | The sample sticker shown below will be affixed to all Level 2A documents | The sample sticker shown below will be affixed to all Level 2B documents |
|---|---|---|
| PERMISSION TO REPRODUCE AND DISSEMINATE THIS MATERIAL HAS BEEN GRANTED BY Sample TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC) 1 | PERMISSION TO REPRODUCE AND DISSEMINATE THIS MATERIAL IN MICROFICHE, AND IN ELECTRONIC MEDIA FOR ERIC COLLECTION SUBSCRIBERS ONLY, HAS BEEN GRANTED BY Sample TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC) 2A | PERMISSION TO REPRODUCE AND DISSEMINATE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY Sample TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC) 2B |
| Level 1 | Level 2A | Level 2B |
| [X] | [ ] | [ ] |
| Check here for Level 1 release, permitting reproduction and dissemination in microfiche or other ERIC archival media (e.g., electronic) *and* paper copy. | Check here for Level 2A release, permitting reproduction and dissemination in microfiche and in electronic media for ERIC archival collection subscribers only | Check here for Level 2B release, permitting reproduction and dissemination in microfiche only |

Documents will be processed as indicated provided reproduction quality permits.
If permission to reproduce is granted, but no box is checked, documents will be processed at Level 1.

*I hereby grant to the Educational Resources Information Center (ERIC) nonexclusive permission to reproduce and disseminate this document as indicated above. Reproduction from the ERIC microfiche or electronic media by persons other than ERIC employees and its system contractors requires permission from the copyright holder. Exception is made for non-profit reproduction by libraries and other service agencies to satisfy information needs of educators in response to discrete inquiries.*

**Sign here, → please**


| | | |
|---|---|---|
| Signature: Richard M Luecht | Printed Name/Position/Title: Richard M. Luecht, Ph.D. Professor & Chair, Edu. Research Methodology | |
| Organization/Address: University of North Carolina at Greensboro Greensboro, NC 27402-6171 | Telephone: 336/334-3473 | FAX: 336/256-0405 |
| | E-Mail Address: rmluecht@uncg.edu | Date: 5/10/00 |


(over)

## III. DOCUMENT AVAILABILITY INFORMATION (FROM NON-ERIC SOURCE):

If permission to reproduce is not granted to ERIC, *or*, if you wish ERIC to cite the availability of the document from another source, please provide the following information regarding the availability of the document. (ERIC will not announce a document unless it is publicly available, and a dependable source can be specified. Contributors should also be aware that ERIC selection criteria are significantly more stringent for documents that cannot be made available through EDRS.)

| |
|---|
| Publisher/Distributor: |
| Address: |
| Price: |

## IV. REFERRAL OF ERIC TO COPYRIGHT/REPRODUCTION RIGHTS HOLDER:

If the right to grant this reproduction release is held by someone other than the addressee, please provide the appropriate name and address:

| |
|---|
| Name: |
| Address: |

## V. WHERE TO SEND THIS FORM:

Send this form to the following ERIC Clearinghouse:

**ERIC CLEARINGHOUSE ON ASSESSMENT AND EVALUATION**
**UNIVERSITY OF MARYLAND**
**1129 SHRIVER LAB**
**COLLEGE PARK, MD 20772**
**ATTN: ACQUISITIONS**

However, if solicited by the ERIC Facility, or if making an unsolicited contribution to ERIC, return this form (and the document being contributed) to:

**ERIC Processing and Reference Facility**
**4483-A Forbes Boulevard**
**Lanham, Maryland 20706**

**Telephone: 301-552-4200**
**Toll Free: 800-799-3742**
**FAX: 301-552-4700**
**e-mail: ericfac@inet.ed.gov**
**WWW: http://ericfac.piccard.csc.com**

EFF-088 (Rev. 2/2000)